DR. HANS-JOACHIM MÜSCHENBORN

To the U. S. Patent and Trademark Office
C/o Examiner Scott M. Collins
Commissioner for Patents
PO Box 1450
Alexandria, Virginia 22313-1450
USA
Tel: +1-703-3057865
Fax: +1-703-8729306

Ihre Zeichen, Ihre Nachricht vom	Unser Zeichen, unsere Nachricht vom	Telefon	Datum
8. July 2004		+41-41-7582962	1. October 2004

CONCERNS: PATENT APPLICATION 09/740,925

Dear Mr. Collins,

Thank you very much for your comments and remarks dated 7/8/2004 regarding Patent Application 09/740,925.

I.

Regarding the Information Disclosure Statements

From our point of view, the non-english citations given in the information disclosure statements are of no particular relevance to the current application.

II.

Regarding the specification

Regarding objections 4-5 the final specification (including all amendments) will be submitted with 1½ line spacing.

Bundesstr. 7 • CH-6304 Zug • Switzerland
Tel: +41-41-7582962 • Fax: +41-41-7582965 • EMail: Dr.Mueschenborn@bluewin.ch

III.

Regarding the drawings

The objections 6-7 are not understood, because it is obvious, that figures in patent applications and/or patents are never self-explanatory and can in allmost all cases only be completely understood together with their textual description.

IV.

Facing the numerous objections (Nos. 8-27) to the claims, the originally filed claims 1 - 36 are canceled.

A new version of independent claim 1 is filed with the current communication.

It is intended to file new dependent claims once the patentability of independent claim 1 has been confirmed.

In the amended version of claim 1 the term "network system" was replaced by the more precice term "interprocess communication system".

In the amended version of claim 1 the term "connection build-up" was replaced by the more precice terms "connection initiation" and/or "connection acceptance".

Compared to originally filed and canceled claim 1 the new claim 1 requires that peripheral threads are being executed only on one or more peripheral units of peripheral units $PE_{1,...,pe}$, i.e. NO peripheral thread is executed on one of the central or service units.

This limits the new claim 1 to interprocess communication systems comprizing

at least 3 different types

of units:

1. at least one Service Unit SE executing at least one Critical Service,
2. at least one Central Unit ZE executing at least one Central Process,
3. at least one Peripheral Unit PE executing at least one peripheral thread.

It is especially emphasised that an interprocess communication system according to the new claim 1 of the current application at **NO time** requires or comprises

1. a physical network connection between a peripheral unit and a Service Unit is required,
2. a direct logical communication connection between a peripheral thread an a critical service.

ARGUMENTS:

The prior art does neither teach nor suggest an Inter Process Communication System of the new independent claim 1:

US 5,699,518 describes only the direct communication between one client node and one server node (see figure 2) and therefore requires a physical network connection between the client node and the server node.

In US 5,699,518 the client node needs to establish a connection to the server node to perfom a RPC call. Even if the top most client program may address servers by logical names, some process and/or thread on the client node has to resolve the logical name of the addressed server at some time into the physical network address (the servers IP-address and port-number and/or MAC-address of the servers network interface) of the unit running the server.

"The client service control manager **determines where to start** the appropriate server code based upon server program provided location context specifications and any location context requested by the client program request. In the case where the determined location ist remote, a server service control manager receives the request forwarded by the clients service control manager and then starts the execution of the server code, if it is not already running. The server code then finds or creates the object or class instance requested by the client program. The server code then marshals an interface of the requested object and sends the marshaled interface back toi the server service control manager. The server service control manager then forwards this marshaled object back to the client program through the client service control manager." (US 5,699,518, col. 5, lines 29-43)

"When required, the client service control manager 202 looks up information in the registration database 203 to determine a desired location context for executing the server code corresponding to the activation request. Also, when required, **the client service control manager** 202 asks the DFS 209 to return a server node network address for given a network path name. **Once the client service control manager 202 knows with which network node to communicate, the client service control manager 202 establishes a remote connection, using standard RPC mechanisms, to the server service control manager 206 on the determined node corresponding to the client program activation request.**" (US 5,699,518, col. 8, line 66 - col. 9, line 10)

"In it's communication, the client service control manager 202 forwards all of the information in the client program activation request to the server service control manager 206." (US 5,699,518, col. 9, lines 14-17)

"Finally, the object server code 205 marshals a pointer to the object instance data structure into a packet and sends the packet to the client program 201 **through the server service control manager 206 and the client service control manager 202**. The client program 201 unmarshals the pointer and is then able to communicate with the server code directly." (US 5,699,518, col. 9, lines 24-30)

Thus, a direct bidirectional logical communication connection is required between the client service control manager and the server service control manager, and hence between at least one thread running on the client node and at least one thread running on the server node.

In the new claim 1 of the current application **NO** peripheral process or peripheral unit ever needs to know the physical network address (the servers IP-address and port-number and/or MAC-address of the servers network interface) of the unit running the Critical Service.

In the new claim 1 of the current application **NO** peripheral process or peripheral unit ever resolves the logical server name into the physical network address (the servers IP-address and port-number and/or MAC-address of the servers network interface) of the unit running the Critical Service.

In the new claim 1 of the current application **NO** direct bidirectional logical communication connection is required between the peripheral thread and the critical service, and hence between threads running on the peripheral unit and threads running on the service unit.

The indirect communication via TWO bidirectional logical communication connections, the first between the peripheral thread and the Central Process and the second between the Central Process and the Critical Service, is the core of the present patent application.

In US 5,699,518 client programs are bound to be executed on the same node as the client service control manager, and remote server code is bound to be executed on the same node as the server service control manager:
"The client program 301 and the client service control manager 306 reside on the client node, and the server service control manager 313 and the object server code 319 reside on the server node." (US 5,699,518, col. 10, lines 4-7)

This is **NOT** the case in new claim 1 of the present application:
Peripheral threads in new claim 1 of the present application do **NEITHER** run on the same unit as the Central Process **NOR** on the same unit as the Critical Service.

In addition, the security context refered to in US 5,699,518 is defined in col. 6 line 62 – col 7 line 13). In explicite, a security context used in US 5,699,518 is typically operating system dependent, and implemented for example as a user account and a password.

The security context described in US 5,699,518 is fundamentally different from the system security established in an interprocess communication system according to the new claim 1, because the security of claim 1 results only from the fact, that their is no technical possibility to establish any further connection to the critical service bejond the explicitely specified connection(s) to the at least one Central Process(es).

Sincerely, Dr. Hans-Joachim Müschenborn